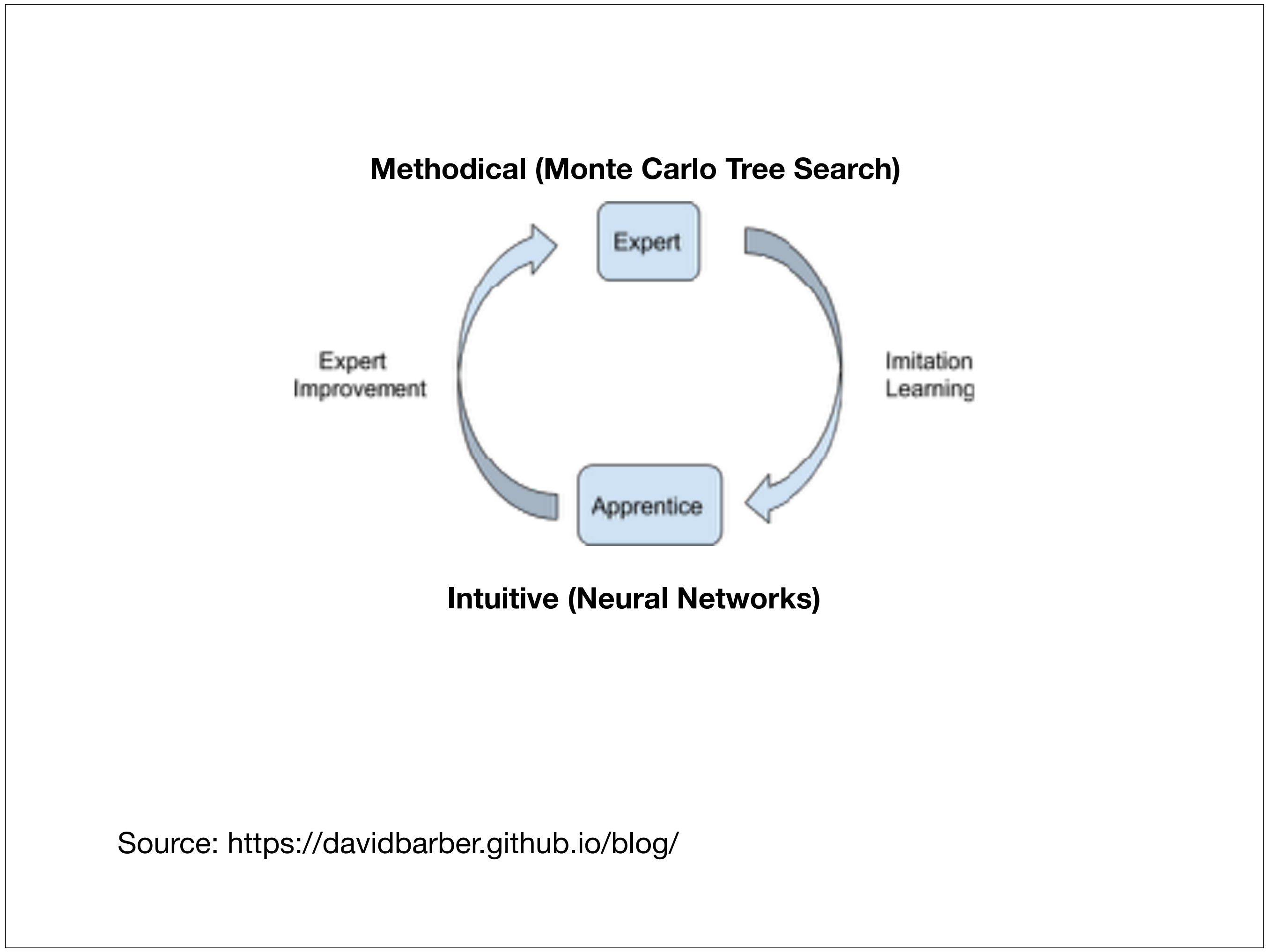
Notes on
“Thinking Fast and Slow with
Deep Learning and Tree Search"¶

Thinking Fast and Slow with Deep Learning and Tree Search by Thomas Anthony, Zheng Tian and David Barber
https://arxiv.org/abs/1705.08439

Notes by Bob Kemp



Methodical (Monte Carlo Tree Search)

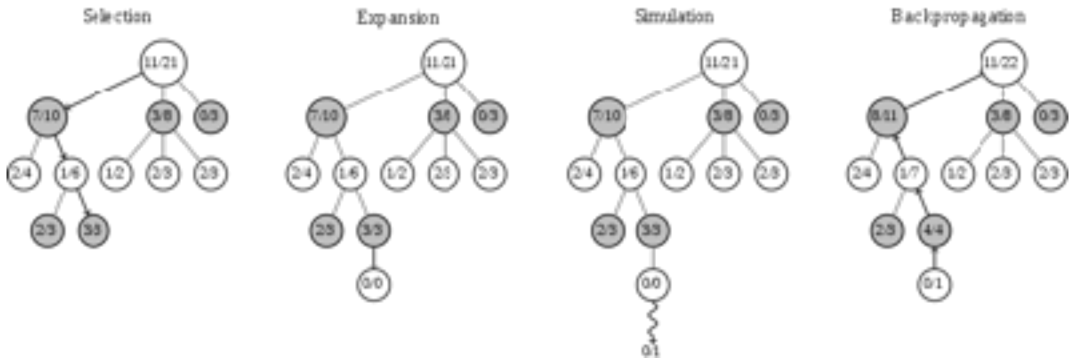
Intuitive (Neural Networks)

Source: https://davidbarber.github.io/blog/

The key idea of this paper is that of an “expert” module training an “apprentice”, which in turn allows the expert to perform better. A virtuous circle, hopefully.

Perhaps better to describe them as “Methodical” and “Intuitive”. MCTS will play through lots of games trying to find winning paths. The NN when given a board state will just return a Win/Lose estimate. MCTS provides the training data for the NN and the NN guides the search for winning moves by MCTS.

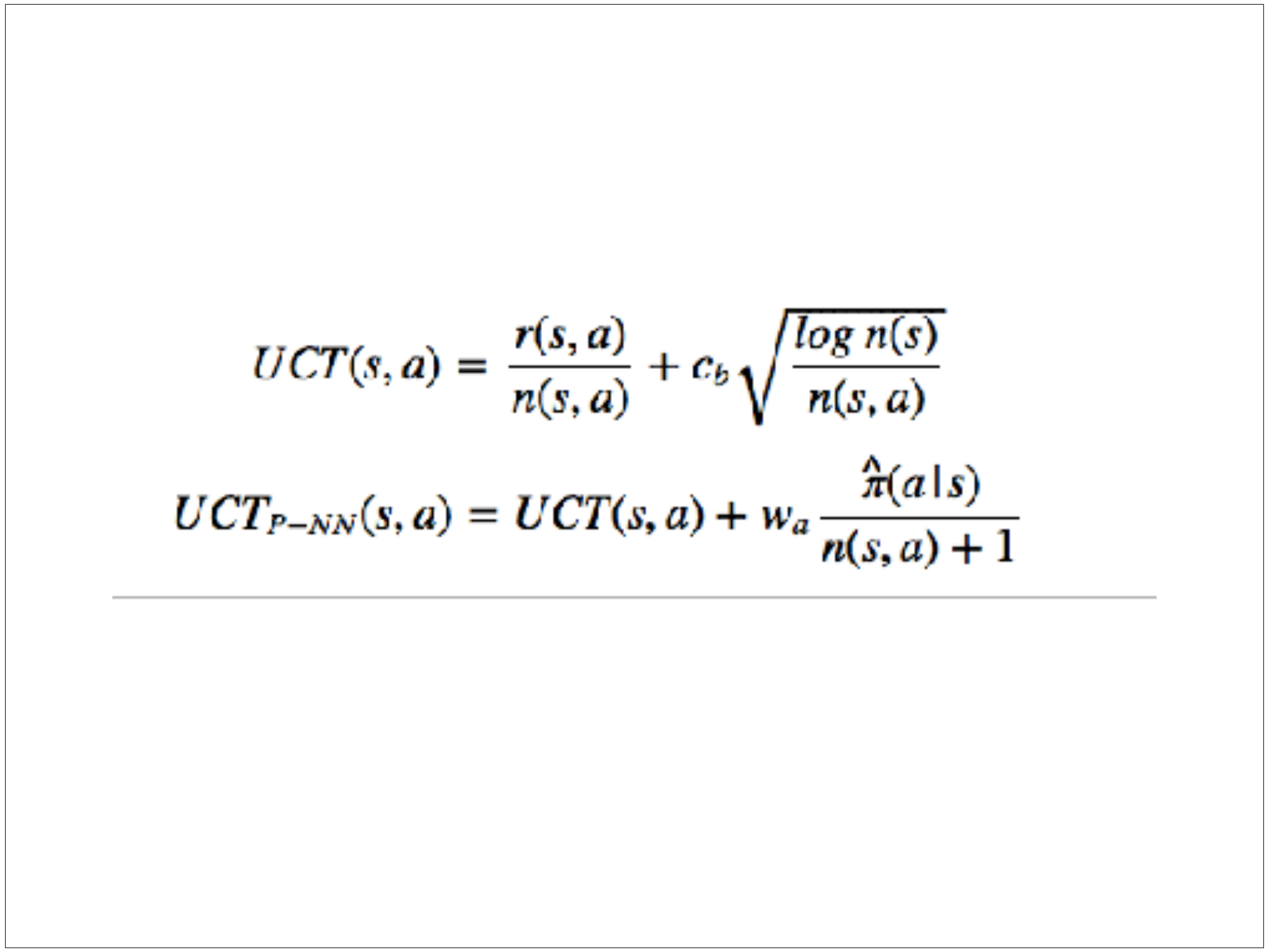
MCTS



1. Tree policy, e.g UCT
2. Counters: total reward + visits
3. Random rollouts
4. Backprop == updating counters

Source: Wikipedia MCTS

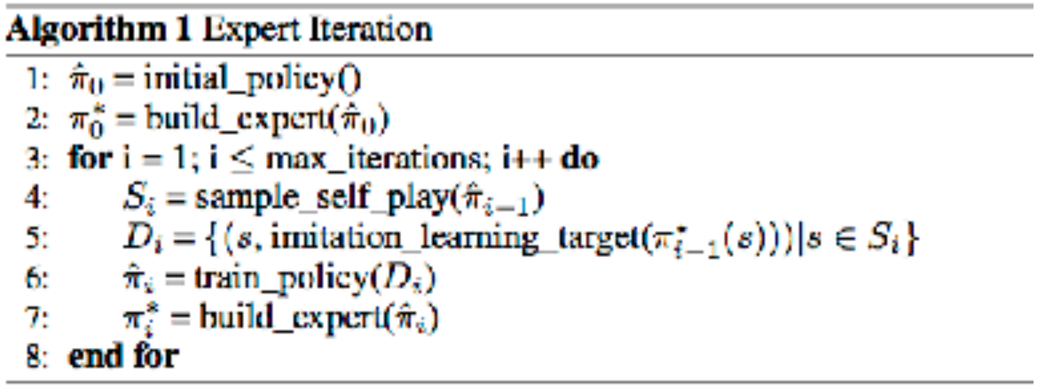
MCTS performs a partial search of the possible game trees guided by a “tree policy”.
Usual policy is UCT which favours unexplored subtrees and those with a high average reward.
Trees include multiple game variants (alternative paths).



$$UCT(s,a) = \frac{r(s,a)}{n(s,a)} + c_b\sqrt{\frac{log\, n(s)}{n(s,a)}}$$

$$UCT_{P-NN}(s,a) = UCT(s,a) + w_a \frac{\hat{\pi}(a|s)}{n(s,a)+1}$$

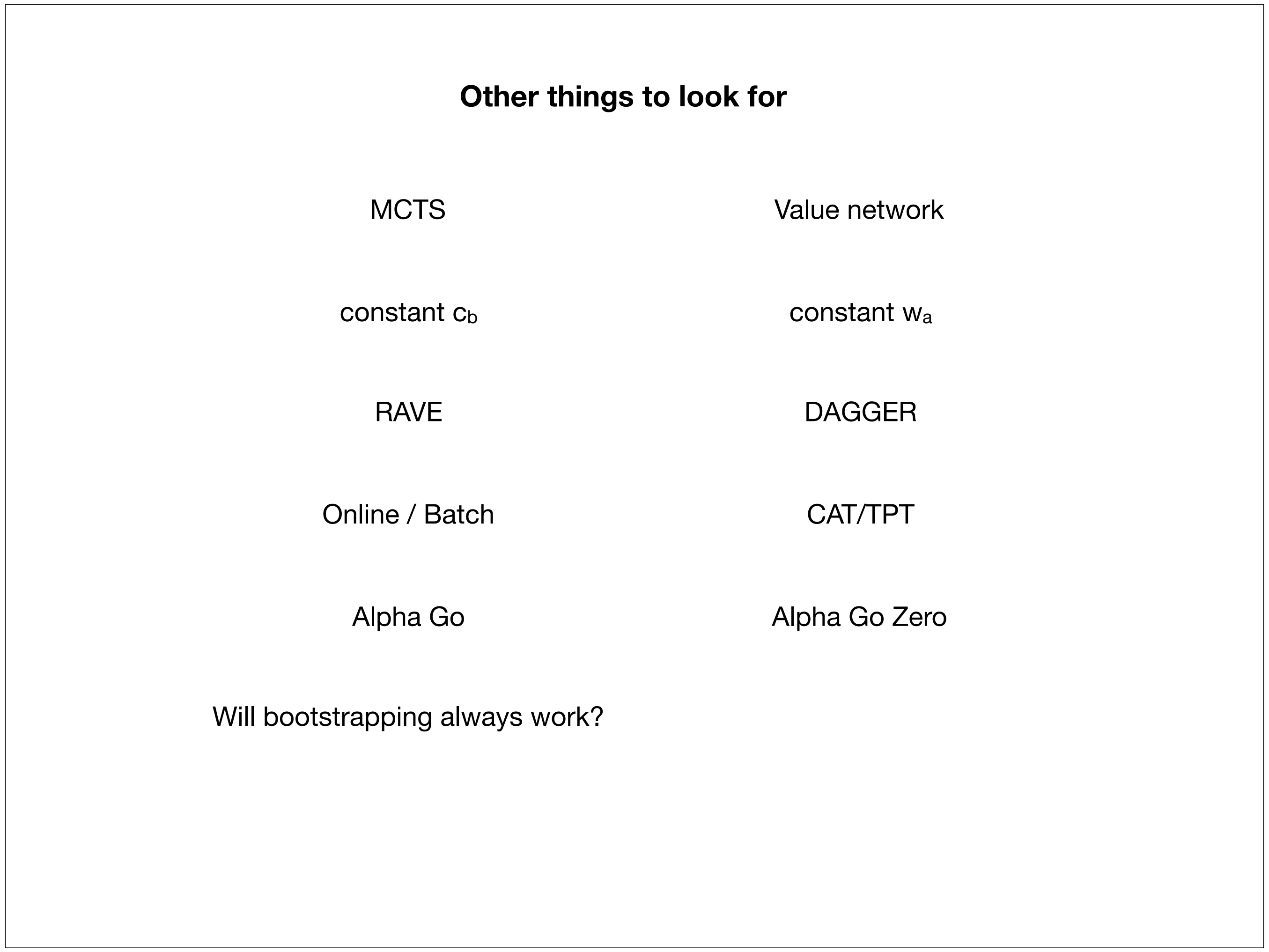
First iteration is vanilla UCT but later ones are influenced by the neural network output.



Algorithm 1 Expert Iteration

```
$\hat{\pi}_0$ = initial_policy()
$\pi_0^*$ = build_expert($\hat{\pi}_0$)
for i = 1; i ≤ max_iterations; i++ do
    $S_i$ = sample_self_play($\hat{\pi}_{i-1}$)
    $D_i = \{(s, \text{imitation\_learning\_target}(\pi^*_{i-1}(s))) | s \in S_i\}$
    $\hat{\pi}_i$ = train_policy($D_i$)
    $\pi_i^*$ = build_expert($\hat{\pi}_i$)
end for
```

Algo 1 is best route into paper. Initial MCTS policy is UCT. Play some games and save 1 board state + outcome for each game. Train network on game data and create augmented policy: UCT + neural network. Loop back using MCTS with new policy.



Other things to look for

MCTS

Value network

constant c_b

constant w_a

RAVE

DAGGER

Online / Batch

CAT/TPT

Alpha Go

Alpha Go Zero

Will bootstrapping always work?

Algorithm 1 is a good way into the paper. You can (e.g.) try to locate each section the paper within the algorithm.

Here also are some things to look for in the paper.